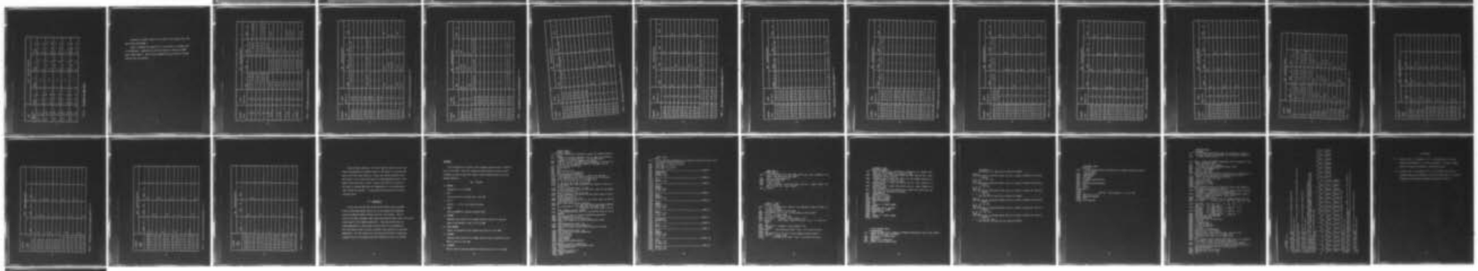
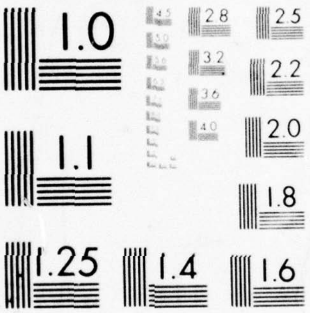
AD-A040 159 SYRACUSE UNIV N Y DEPT OF INDUSTRIAL ENGINEERING AND--ETC F/G 9/2
RANDOM ACCESS MEMORY FAULT LOCATION CAPABILITY OF THE ALGORITHM--ETC(U)
MAY 77 J KNAIZUK, C R HARTMANN F30602-71-C-0312

UNCLASSIFIED TR-76-10 RADC-TR-77-150 NL

1 OF 1
AD
A040159



END
DATE
FILMED
6-77


1.0
4.5
5.0
5.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

(11)

RADC-TR-77-150
Final Technical Report
May 1977

RANDOM ACCESS MEMORY FAULT LOCATION CAPABILITY OF THE ALGORITHM TEST SEQUENCE

Syracuse University

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED: Anthony Coppola
ANTHONY COPPOLA
Project Engineer

APPROVED: Joseph J Naresky
JOSEPH J. NARESKY
Chief, Reliability & Compatibility Division

FOR THE COMMANDER: John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

(14) TR-76-10

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) **REPORT DOCUMENTATION PAGE**		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) RADC-TR-77-150	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) RANDOM ACCESS MEMORY FAULT LOCATION CAPABILITY OF THE ALGORITHM TEST SEQUENCE.		5. TYPE OF REPORT & PERIOD COVERED (9) Final Technical Report. Jun[redacted] - to Oct[redacted] 1976,
		6. PERFORMING ORG. REPORT NUMBER 76-10
7. AUTHOR(s) (10) J. Knaizuk, Jr. C. R. P. Hartmann		8. CONTRACT OR GRANT NUMBER(s) (15) F30602-71-C-0312
9. PERFORMING ORGANIZATION NAME AND ADDRESS Syracuse University Department of Industrial Engineering + Opns. Res. Syracuse NY 13210		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 45400526
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBRT) Griffiss AFB NY 13441		12. REPORT DATE (11) May [redacted]977
		13. NUMBER OF PAGES 44 (12) 41p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same (16) 4540 (17) 05		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Anthony Coppola (RBRT)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Reliability Random Access Memory Fault Location Testing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes an Algorithm Test Sequence (ATS) designed to provide an optimal test sequence for detecting stuck-at faults in a random access memory. An analysis of its capability to locate any single stuck-at fault is provided.		

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 184 mc

EVALUATION

The algorithm investigated in this volume will be of immediate use in memory testing, particularly as a preliminary test to verify the existence of each bit in a memory prior to testing for pattern sensitivities. It should be pointed out that the assumption of noncreative decoders may not be reasonable without an additional set of output test points, since while inverted inputs are not used, the example given immediately provides all inverted outputs, and if one of the inverters fails, the decoder will become creative.

Anthony Coppola

ANTHONY COPPOLA
Project Engineer

I. INTRODUCTION

This paper is a continuation of the analysis of an optimal test sequence for detecting stuck-at (s-a) faults in a Random Access Memory (RAM). The past two papers (1), (2) on the Algorithm Test Sequence (ATS) have investigated the ability of the ATS to detect s-a faults. The first paper (1) deals with any single s-a-0 or s-a-1 fault while the second paper (2) shows the ability of the ATS to detect any combination of multiple s-a-0 and s-a-1 faults in a RAM. In this paper we will analyze the capability of the ATS to diagnose (locate) any single s-a fault.

Each subsystem of a RAM will be analyzed and to provide insight a computer program to simulate a 32 x 1 bit RAM with induced errors will be tested by another computer program.

For completness we will again present the ATS.

II. ATS - An Optimal RAM Test Algorithm

In order to obtain greater diagnostic capability from the ATS the sequence of operations in the last steps have been permutated. The new sequence is given below and then summarized in Figure 1.

Before stating the algorithm let us introduce some notations.

Let A_μ be the memory address μ,

$0 \leq \mu < 2^n$.

Let:

$$\pi_0 = [A_\mu \mid \mu \equiv 0 (\text{modulo } 3)]$$
$$\pi_1 = [A_\mu \mid \mu \equiv 1 (\text{modulo } 3)]$$
$$\pi_2 = [A_\mu \mid \mu \equiv 2 (\text{modulo } 3)]$$

Algorithm:

Step 1 Write the all-0 word, W_0, at all locations

$A_j \in \pi_1$ and $A_k \in \pi_2$

Step 2 Write the all-1 word, W_1, at all locations

$A_i \in \pi_0$

Step 3 Read all locations $A_j \in \pi_1$

$$\text{If output} \begin{cases} = W_0\text{; no fault indicated} \\ \neq W_0\text{; RAM fault indicated} \end{cases}$$

Step 4 Write the all-1 word, W_1, at all locations

$A_j \in \pi_1$

Step 5 Read all locations $A_k \varepsilon \pi_2$

$$\text{If output} \begin{cases} = W_0; \text{ no fault indicated} \\ \neq W_0; \text{ RAM fault indicated} \end{cases}$$

Step 6 Read all locations $A_i \varepsilon \pi_0$

$$\text{If output} \begin{cases} = W_1; \text{ no fault indicated} \\ \neq W_1; \text{ RAM fault indicated} \end{cases}$$

Step 7 Write the all 0 word, W_0, at all locations

$A_i \varepsilon \pi_0$

Step 8 Write the all 1 word, W_1, at all locations

$A_k \varepsilon \pi_2$

Step 9 Read the all 0 word, W_0, at all locations

$A_i \varepsilon \pi_0$

$$\text{If output} \begin{cases} = W_0; \text{ no fault indicated} \\ \neq W_0; \text{ RAM fault indicated} \end{cases}$$

Step 10 Read the all 1 word, W_1, at all locations

$A_k \varepsilon \pi_2$

$$\text{If output} \begin{cases} = W_1; \text{ no fault indicated} \\ \neq W_1; \text{ RAM fault indicated} \end{cases}$$

Step 11 Read the all 1 word, W_1 at all locations

$A_j \varepsilon \pi_1$

$$\text{If output} \begin{cases} = W_1; \text{ no fault indicated} \\ \neq W_1; \text{ RAM fault indicated} \end{cases}$$

END.

The above sequence requires a total of 4×2^n memory accesses.

Memory Array Addresses	ATS Step										
	1	2	3	4	5	6	7	8	9	10	11
π_o		Wr W_1				Rd W_1	Wr W_0		Rd W_0		
π_1	Wr W_0		Rd W_0	Wr W_1							Rd W_1
π_2	Wr W_0				Rd W_0			Wr W_1		Rd W_1	

Where Wr = Write; Rd = Read

Figure 1: ATS

VI. Decoder Single Stuck-At Faults

Again we will use an APL program to simulate the behavior of single stuck-at faults in the Decoder. The test results for faults on the Decoder inputs are not shown here since they are indistinguishable from the faults in the MAR as given in Table 1. The faults within the decoder are simulated by the APL program AUTODEC and the results are tabulated in Tables 2, 3 and 4. The results for a s-a-1 output on any single AND gate in a non-creative Decoder (1) were the same. That is, if the output of an AND gate is s-a-1 then the ATS will give the same results no matter what AND gate is stuck. The results of this type of s-a-1 fault is shown in Table 2.

Fault	ATS		
	Step 3	Step 5	Step 9
Any Decoder AND Gate Output s-a-1	π_1	π_2	π_0

Table 2: ATS Output for Decoder Faults

III. MDR Single Stuck-At Fault

If the MDR contains any stuck-at bit then this bit position will remain stuck for all memory locations accessed by the ATS. This would signify that the MDR is at fault and the stuck-at bit is readily found. Note that the stuck-at bit in the MDR is found after the second wrong reading from the RAM due to our single fault assumption.

IV. Memory Array Single Stuck-At Fault

For a single stuck-at bit in a memory array word whose size is greater than one bit, diagnosis by the ATS is obtained when the erroneous word is read after or before a correct reading is obtained from the RAM.

For memory arrays with only one bit words the diagnosis for a single stuck-at fault in a word is not separated from a Memory Address Register (MAR) or a Decoder fault until the first five steps of the ATS are finished.

If any one memory array word reads wrong within the first five steps or at any point after the first five steps of the ATS, the memory array word is at fault. Otherwise the fault is in the MAR or the Decoder.

V. MAR Single Stuck-At Faults

In order to show the effects of single faults within the MAR (and Decoder) a 32 x 1 bit RAM was simulated with an APL program. Another APL program was used to simulate all single s-a MAR faults. These programs may be found in the Appendix of this report. The output of the ATS program for the simulated faults has been reduced to the tables that will be found in what follows.

Table 1 shows the results of the computer output for single s-a faults in a five bit MAR.

As seen from Table 1 the output of the ATS is capable of locating which bit is stuck but unable to determine if the bit is s-a-0 or s-a-1.

The only problem which exists is the inability of the ATS to distinguish between a MAR fault and a fault on the input lines of the Decoder (1), (2).

MAR	ATS Detected Failures			
Stuck-At Fault	Step 3	Step 5	Step 9	Step 11
x_0 s-a-0 or s-a-1	1 7 13 19 25 31	π_2	3 9 15 21 27	1 7 13 19 25 31
x_1 s-a-0 or s-a-1	1 4 13 16 25 28	π_2	0 9 12 21 24	1 4 13 16 25 28
x_2 s-a-0 or s-a-1	4 7 13 22 28 31	π_2	6 12 15 21 30	4 7 13 22 28 31
x_3 s-a-0 or s-a-1	1 4 7 16 19 22	π_2	0 3 6 18 21	1 4 7 16 19 22
x_4 s-a-0 or s-a-1	16 19 22 25 28 31	π_2	18 21 24 27 30	16 19 22 25 28 31

Table 1: ATS Output for MAR Faults

Therefore we cannot locate a s-a-1 fault on the output of an AND gate within the Decoder.

Table 3 presents the results for a s-a-1 fault on a single input to an AND gate. Choosing a s-a-0 on any input or output of an AND gate yields Table 4. Here we have assumed that W_0 is read by the MDR when no line is accessed.

Decoder Gate	Input line s-a-1	ATS Detected Failure			
		Step 3	Step 5	Step 9	Step 11
$\bar{x}_4$	$\bar{x}_4$	16 19 22 25 28 31		18 21 24 27 30	
$\bar{x}_3$	$\bar{x}_3$	1 4 7 16 19 22		0 3 6 18 21	1 4 7 16 19 22
$\bar{x}_2$	$\bar{x}_2$	4 7 13 22 28 31		6 12 15 21 30	
$\bar{x}_1$	$\bar{x}_1$	1 4 13 16 25 28		0 9 12 21 24	1 4 13 16 25 28
$\bar{x}_0$	$\bar{x}_0$	1 7 13 19 25 31		3 9 15 21 27	
$x_1\ x_0$	x_0	7 19 31	2 11 14 23 26	3 15 27	7 19 31
	x_1	1 13 25	5 11 17 23 29	9 21	
$x_1\ \bar{x}_0$	$\bar{x}_0$	7 19 31	2 11 14 23 26	3 15 27	
	x_1	4 16 28	2 8 14 20 26	0 12 24	
$\bar{x}_1\ x_0$	x_0	1 13 25	5 8 17 20 29	9 21	1 13 25
	$\bar{x}_1$	1 13 25	5 11 17 23 29	9 21	1 13 25
$\bar{x}_1\ \bar{x}_0$	$\bar{x}_0$	1 13 25	5 8 17 20 29	9 21	
	$\bar{x}_1$	4 16 28	2 8 14 20 26	0 12 24	4 16 18

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder	Input	ATS Detected Failure			
Gate	line s-a-1	Step 3	Step 5	Step 9	Step 11
$x_2\ x_1\ x_0$	$x_1\ x_0$	4 7 13 22 28 31	5 14 20 23 29	6 12 15 21 30	
$x_2\ x_1\ \bar{x}_0$	$x_1\ \bar{x}_0$	"	"	"	
$x_2\ \bar{x}_1\ x_0$	$\bar{x}_1\ x_0$	"	"	"	
$x_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_1\ \bar{x}_0$	"	"	"	
$\bar{x}_2\ x_1\ x_0$	$x_1\ x_0$	1 10 16 19 25	2 8 11 17 26	0 3 9 18 24 27	
$\bar{x}_2\ x_1\ \bar{x}_0$	$x_1\ \bar{x}_0$	"	"	"	
$\bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_1\ x_0$	"	"	"	
$\bar{x}_1\ \bar{x}_2\ \bar{x}_0$	$\bar{x}_1\ \bar{x}_0$	"	"	"	
$x_2\ x_1\ x_0$	x_2	7 31	11 13	15	7 31
$x_2\ x_1\ \bar{x}_0$	x_2	22	2 14 26	6 30	22
$x_2\ \bar{x}_1\ x_0$	x_2	13	5 17 29	21	13
$x_2\ \bar{x}_1\ \bar{x}_0$	x_2	4 28	8 20	12	4 28
$\bar{x}_2\ x_1\ x_0$	$\bar{x}_2$	7 31	11 13	15	

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder Gate	Input line s-a-1	ATS Detected Failure			
		Step 3	Step 5	Step 9	Step 11
$\bar{x}_2\ x_1\ \bar{x}_0$	$\bar{x}_2$	22	2 14 26	6 30	
$\bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_2$	13	5 17 29	21	
$\bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_2$	4 28	8 20	12	
$x_3\ x_2\ x_1\ x_0$	$x_2\ x_1\ x_0$	10 13 25 28 31	8 11 14 26 29	9 12 15 24 27 30	
$x_3\ x_2\ x_1\ \bar{x}_0$	$x_2\ x_1\ \bar{x}_0$	"	"	"	
$x_3\ x_2\ \bar{x}_1\ x_0$	$x_2\ \bar{x}_1\ x_0$	"	"	"	
$x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	$x_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$x_3\ \bar{x}_2\ x_1\ x_0$	$\bar{x}_2\ x_1\ x_0$	"	"	"	
$x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$\bar{x}_2\ x_1\ \bar{x}_0$	"	"	"	
$x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_2\ \bar{x}_1\ x_0$	"	"	"	
$x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$\bar{x}_3\ x_2\ x_1\ x_0$	$x_2\ x_1\ x_0$	1 4 7 16 19 22	2 15 17 20 23	0 3 6 18 21	
$\bar{x}_3\ x_2\ x_1\ \bar{x}_0$	$x_2\ x_1\ \bar{x}_0$	"	"	"	

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder Gate	Input line s-a-1	ATS Detected Failure			
		Step 3	Step 5	Step 9	Step 11
$\bar{x}_3\ x_2\ \bar{x}_1\ x_0$	$x_2\ \bar{x}_1\ x_0$	1 4 7 16 19 22	2 15 17 20 23	0 3 6 18 21	
$\bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$	$x_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$\bar{x}_3\ \bar{x}_2\ x_1\ x_0$	$\bar{x}_2\ x_1\ x_0$	"	"	"	
$\bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$\bar{x}_2\ x_1\ \bar{x}_0$	"	"	"	
$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_2\ \bar{x}_1\ x_0$	"	"	"	
$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$x_3\ x_2\ x_1\ x_0$	x_3	7	23		
$x_3\ x_2\ x_1\ \bar{x}_0$	x_3	22	14	6	
$x_3\ x_2\ \bar{x}_1\ x_0$	x_3		5 29	21	
$x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	x_3	4	20		
$x_3\ \bar{x}_2\ x_1\ x_0$	x_3	19	11	3	
$x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	x_3		2 26	18	
$x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	x_3	1	17		

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder Gate	Input line s-a-1	ATS Detected Failure			
		Step 3	Step 5	Step 9	Step 11
$x_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$	x_3	16	8	0	
$\bar{x}_3 x_2 x_1 x_0$	$\bar{x}_3$	7	23		7
$\bar{x}_3 x_2 x_1 \bar{x}_0$	$\bar{x}_3$	22	14	6	22
$\bar{x}_3 x_2 \bar{x}_1 x_0$	$\bar{x}_3$		5 29	21	
$\bar{x}_3 x_2 \bar{x}_1 \bar{x}_0$	$\bar{x}_3$	4	20		4
$\bar{x}_3 \bar{x}_2 x_1 x_0$	$\bar{x}_3$	19	11	3	19
$\bar{x}_3 \bar{x}_2 x_1 \bar{x}_0$	$\bar{x}_3$		2 26	18	
$\bar{x}_3 \bar{x}_2 \bar{x}_1 x_0$	$\bar{x}_3$	1	17		1
$\bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$	$\bar{x}_3$	16	8	0	16
$x_4 x_3 x_2 x_1 x_0$	$x_3 x_2 x_1 x_0$	16 19 22 25 28 31	17 20 23 26 29	18 21 24 27 30	
$x_4 x_3 x_2 x_1 \bar{x}_0$	$x_3 x_2 x_1 \bar{x}_0$	"	"	"	
$x_4 x_3 x_2 \bar{x}_1 x_0$	$x_3 x_2 \bar{x}_1 x_0$	"	"	"	
$x_4 x_3 x_2 \bar{x}_1 \bar{x}_0$	$x_3 x_2 \bar{x}_1 \bar{x}_0$	"	"	"	

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder Gate	Input line s-a-1	ATS Detected Failure Step 3	 Step 5	 Step 9	 Step 11
$x_4\ x_3\ \bar{x}_2\ x_1\ x_0$	$x_3\ \bar{x}_2\ x_1\ x_0$	16 19 22 25 28 31	17 20 23 26 29	18 21 24 27 30	
$x_4\ x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	"	"	"	
$x_4\ x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	"	"	"	
$x_4\ x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$x_4\ \bar{x}_3\ x_2\ x_1\ x_0$	$\bar{x}_3\ x_2\ x_1\ x_0$	"	"	"	
$x_4\ \bar{x}_3\ x_2\ x_1\ \bar{x}_0$	$\bar{x}_3\ x_2\ x_1\ \bar{x}_0$	"	"	"	
$x_4\ \bar{x}_3\ x_2\ \bar{x}_1\ x_0$	$\bar{x}_3\ x_2\ \bar{x}_1\ x_0$	"	"	"	
$x_4\ \bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$x_4\ \bar{x}_3\ \bar{x}_2\ x_1\ x_0$	$\bar{x}_3\ \bar{x}_2\ x_1\ x_0$	"	"	"	
$x_4\ \bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$\bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	"	"	"	
$x_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$	"	"	"	
$x_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$\bar{x}_4\ x_3\ x_2\ x_1\ x_0$	$x_3\ x_2\ x_1\ x_0$	1 4 7 10 13	2 5 8 11 14	0 3 6 9 12 15	

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder	Input	ATS Detected Failure			
Gate	line s-a-1	Step 3	Step 5	Step 9	Step 11
$\bar{x}_4\ x_3\ x_2\ x_1\ \bar{x}_0$	$x_3\ x_2\ x_1\ \bar{x}_0$	1 4 7 10 13	2 5 8 11 14	0 3 6 9 12 15	
$\bar{x}_4\ x_3\ x_2\ \bar{x}_1\ x_0$	$x_3\ x_2\ \bar{x}_1\ x_0$	"	"	"	
$\bar{x}_4\ x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	$x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$\bar{x}_4\ x_3\ \bar{x}_2\ x_1\ x_0$	$x_3\ \bar{x}_2\ x_1\ x_0$	"	"	"	
$\bar{x}_4\ x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	"	"	"	
$\bar{x}_4\ x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	"	"	"	
$\bar{x}_4\ x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$\bar{x}_4\ \bar{x}_3\ x_2\ x_1\ x_0$	$\bar{x}_3\ x_2\ x_1\ x_0$	"	"	"	
$\bar{x}_4\ \bar{x}_3\ x_2\ x_1\ \bar{x}_0$	$\bar{x}_3\ x_2\ x_1\ \bar{x}_0$	"	"	"	
$\bar{x}_4\ \bar{x}_3\ x_2\ \bar{x}_1\ x_0$	$\bar{x}_3\ x_2\ \bar{x}_1\ x_0$	"	"	"	
$\bar{x}_4\ \bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ x_1\ x_0$	$\bar{x}_3\ \bar{x}_2\ x_1\ x_0$	"	"	"	
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$\bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	"	"	"	

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder Gate	Input line s-a-1	ATS Detected Failure			
		Step 3	Step 5	Step 9	Step 11
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$	1 4 7 10 13	2 5 8 11 14	0 3 6 9 12 15	
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	"	"	"	
$x_4\ x_3\ x_2\ x_1\ x_0$	x_4	31			31
$x_4\ x_3\ x_2\ x_1\ \bar{x}_0$	x_4		14	30	
$x_4\ x_3\ x_2\ \bar{x}_1\ x_0$	x_4		29		
$x_4\ x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	x_4	28			28
$x_4\ x_3\ \bar{x}_2\ x_1\ x_0$	x_4		11	27	
$x_4\ x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	x_4		26		
$x_4\ x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	x_4	25			25
$x_4\ x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	x_4		8	24	
$x_4\ \bar{x}_3\ x_2\ x_1\ x_0$	x_4		23		
$x_4\ \bar{x}_3\ x_2\ x_1\ \bar{x}_0$	x_4	22			22
$x_4\ \bar{x}_3\ x_2\ \bar{x}_1\ x_0$	x_4		5	21	

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder Gate	Input line s-a-1	ATS Detected Failure			
		Step 3	Step 5	Step 9	Step 11
$x_4\ \bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$	x_4		20		
$x_4\ \bar{x}_3\ \bar{x}_2\ x_1\ x_0$	x_4	19			19
$x_4\ \bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	x_4		2	18	
$x_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$	x_4		17		
$x_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	x_4	16			16
$\bar{x}_4\ x_3\ x_2\ x_1\ x_0$	$\bar{x}_4$	31			
$\bar{x}_4\ x_3\ x_2\ x_1\ \bar{x}_0$	$\bar{x}_4$		14	30	
$\bar{x}_4\ x_3\ x_2\ \bar{x}_1\ x_0$	$\bar{x}_4$		29		
$\bar{x}_4\ x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_4$	28			
$\bar{x}_4\ x_3\ \bar{x}_2\ x_1\ x_0$	$\bar{x}_4$		11	27	
$\bar{x}_4\ x_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$\bar{x}_4$		26		
$\bar{x}_4\ x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_4$	25			
$\bar{x}_4\ x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_4$		8	24	

TABLE 3: ATS Output for s-a-1 Decoder Faults (con't)

Decoder Gate	Input line s-a-1	ATS Detected Failure Step 3	Step 5	Step 9	Step 11
$\bar{x}_4\ \bar{x}_3\ x_2\ x_1\ x_0$	$\bar{x}_4$		23		
$\bar{x}_4\ \bar{x}_3\ x_2\ x_1\ \bar{x}_0$	$\bar{x}_4$	22			
$\bar{x}_4\ \bar{x}_3\ x_2\ \bar{x}_1\ x_0$	$\bar{x}_4$		5	21	
$\bar{x}_4\ \bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_4$		20		
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ x_1\ x_0$	$\bar{x}_4$	19			
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	$\bar{x}_4$		2	18	
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$	$\bar{x}_4$		17		
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	$\bar{x}_4$	16			

TABLE 3: ATS Output for s-a-1 Decoder Faults

Decoder	ATS	Detected Failures		
Gate s-a-0	Step 6	Step 10	Step 11	
$\bar{x}_4$	0 3 6 9 12 15	2 5 8 11 14	1 4 7 10 13	
$\bar{x}_3$	0 3 6 18 21	2 5 17 20 23	1 4 7 16 19 22	
$\bar{x}_2$	0 3 9 18 24 27	2 8 11 17 26	1 10 16 25 28	
$\bar{x}_1$	0 9 12 21 24	5 8 17 20 29	1 4 13 16 25 28	
$\bar{x}_0$	0 6 12 18 24 30	2 8 14 20 26	4 10 16 22 28	
x_1 x_0	3 15 27	11 23	7 9 31	
x_1 $\bar{x}_0$	6 18 30	2 14 26	10 32	
$\bar{x}_1$ x_0	9 21	5 17 29	1 13 25	
$\bar{x}_1$ $\bar{x}_0$	0 12 24	8 20	4 16 28	
x_2 x_1 x_0	15	23	7 31	
x_2 x_1 $\bar{x}_0$	6 30	14	22	
x_2 $\bar{x}_1$ x_0	21	5 29	13	
x_2 $\bar{x}_1$ $\bar{x}_0$	12	20	4 28	

TABLE 4: ATS Output for s-a-0 Decoder Faults (con't)

Decoder Gate s-a-0	ATS Detected Failures			
	Step 6	Step 10	Step 11	
$\bar{x}_2\ x_1\ x_0$	3 27	11	19	
$\bar{x}_2\ x_1\ \bar{x}_0$	18	2 26	10	
$\bar{x}_2\ \bar{x}_1\ x_0$	9	17	1 25	
$\bar{x}_2\ \bar{x}_1\ \bar{x}_0$	0 24	8	16	
$x_3\ x_2\ x_1\ x_0$	15		31	
$x_3\ x_2\ x_1\ \bar{x}_0$	30	14		
$x_3\ x_2\ \bar{x}_1\ x_0$		29	13	
$x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	12		28	
$x_3\ \bar{x}_2\ x_1\ x_0$	27	11		
$x_3\ \bar{x}_2\ x_1\ \bar{x}_0$		26	10	
$x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	9		25	
$x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	24	8		
$\bar{x}_3\ x_2\ x_1\ x_0$		23	7	

TABLE 4: ATS Output for s-a-0 Decoder Faults (con't)

Decoder Gate s-a-0	ATS	Detected Failures		
	Step 6	Step 10	Step 11	
$\bar{x}_3\ x_2\ x_1\ \bar{x}_0$	6		22	
$\bar{x}_3\ x_2\ \bar{x}_1\ x_0$	21	5		
$\bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$		20	4	
$\bar{x}_3\ \bar{x}_2\ x_1\ x_0$	3		19	
$\bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	18	2		
$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$		17	1	
$\bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	0	16		
$x_4\ x_3\ x_2\ x_1\ x_0$			31	
$x_4\ x_3\ x_2\ x_1\ \bar{x}_0$	30			
$x_4\ x_3\ x_2\ \bar{x}_1\ x_0$		29		
$x_4\ x_3\ x_2\ \bar{x}_1\ \bar{x}_0$			28	
$x_4\ x_3\ \bar{x}_2\ x_1\ x_0$	27			
$x_4\ x_3\ \bar{x}_2\ x_1\ \bar{x}_0$		26		

TABLE 4: ATS Output for s-a-0 Decoder Faults (con't)

Decoder Gate s-a-0	ATS	Detected Failures		
	Step 6	Step 10	Step 11	
$x_4\ x_3\ \bar{x}_2\ \bar{x}_1\ x_0$			25	
$x_4\ x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	24			
$x_4\ \bar{x}_3\ x_2\ x_1\ x_0$		23		
$x_4\ \bar{x}_3\ x_2\ x_1\ \bar{x}_0$			22	
$x_4\ \bar{x}_3\ x_2\ \bar{x}_1\ x_0$	21			
$x_4\ \bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$		20		
$x_4\ \bar{x}_3\ \bar{x}_2\ x_1\ x_0$			19	
$x_4\ \bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$	18			
$x_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$		17		
$x_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$			16	
$\bar{x}_4\ x_3\ x_2\ x_1\ x_0$	15			
$\bar{x}_4\ x_3\ x_2\ x_1\ \bar{x}_0$		14		
$\bar{x}_4\ x_3\ x_2\ \bar{x}_1\ x_0$			13	

TABLE 4: ATS Output for s-a-0 Decoder Faults (con't)

Decoder Gate s-a-0	ATS Detected Failures			
	Step 6	Step 10	Step 11	
$\bar{x}_4\ x_3\ x_2\ \bar{x}_1\ \bar{x}_0$	12			
$\bar{x}_4\ x_3\ \bar{x}_2\ x_1\ x_0$		11		
$\bar{x}_4\ x_3\ \bar{x}_2\ x_1\ \bar{x}_0$			10	
$\bar{x}_4\ x_3\ \bar{x}_2\ \bar{x}_1\ x_0$	9			
$\bar{x}_4\ x_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$		8		
$\bar{x}_4\ \bar{x}_3\ x_2\ x_1\ x_0$			7	
$\bar{x}_4\ \bar{x}_3\ x_2\ x_1\ \bar{x}_0$	6			
$\bar{x}_4\ \bar{x}_3\ x_2\ \bar{x}_1\ x_0$		5		
$\bar{x}_4\ \bar{x}_3\ x_2\ \bar{x}_1\ \bar{x}_0$			4	
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ x_1\ x_0$	3			
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ x_1\ \bar{x}_0$		2		
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ x_0$			1	
$\bar{x}_4\ \bar{x}_3\ \bar{x}_2\ \bar{x}_1\ \bar{x}_0$	0			

TABLE 4: ATS Output for s-a-0 Decoder Faults

From the above tables we can first conclude that any s-a-0 fault within the Decoder is locatable except if the output of a Decoder AND gate on the last level (that is, a gate that chooses a Memory Array word line) is s-a-0 then this fault is indistinguishable from a 1-bit Memory Array word that is s-a-0. Next we see that s-a-1 faults on the input of Decoder AND gates are diagnosible to a particular gate level within the decoder. In some cases we may be able to find the erroneous gate.

VII CONCLUSION

We see that the ATS can locate certain faults within the RAM. Certain indistinguishable faults can not be possibly distinguished unless the RAM has added circuitry just for this purpose. This is true for the MAR vs Decoder input faults and Decoder Memory Array word select line output vs 1-bit Memory Array Word. The other faults that are indistinguishable in the Decoder could be found if we expanded the ATS, but then we would not have an optimal test sequence for detecting RAM faults. The best method for alleviating the problem of indistinguishable faults is by adding some self-checking circuitry to the RAM.

APPENDIX

The following is a listing of APL computer programs used to simulate a 32 x 1 bit RAM. After the listing of some programs a partial output is shown in order to give the reader a better understanding of how the program operates.

APL Programs

1. MEMORY

Simulate a 32 x 1 bit RAM

2. ATS

Simulates the ATS on above 32 x 1 bit RAM

3. OUT

Used by the ATS to list detected faults

4. PRINT

Used by AUTODEC to indicate induced fault

5. AUTODEC

Generates the tests for all single stuck-at faults of a specific type in the Decoder of the 32 x 1 bit RAM

6. CLEAR ERRORS

Clears all previous errors induced into the 32 x 1 bit RAM

7. AUTOMAR

Generates and tests for all single stuck-at faults induced into the MAR of the 32 x 1 bit RAM

8. SETERROR

May be used to place any specific errors into the 32 x 1 bit RAM.

```
     ∇MEMORY [⎕]∇
   ∇ MEMORY ADDRESS
[1]   ⍝ THIS PROGRAM WILL SIMULATE A 32×1 BIT RANDOM ACCESS M
      EMORY.
[2]   ⍝ RDWR IS A GLOBAL VARIABLE THAT IS USED TO INDICATE IF
       A READ OR WRITE OPERATION IS TO BE PERFORMED.
[3]   ⍝ RDWR←1 IS A READ ; RDWR←0 IS A WRITE.
[4]   ⍝ IN IS A GLOBAL VARIABLE WHICH REPRESENTS THE INPUT TO
       THE MDR (OR THE REQUIRED OUTPUT). EITHTER 1 OR 0.
[5]   OUTPUT←STKINX←NUMACS←⍴0
[6]   SAD←⍴ADDRESS
[7]   K←0
[8]  LOOP:→0×⍳SAD<K←K+1
[9]   MAR← 2 2 2 2 2 ⊤ADDRESS[K]
[10]  MAR←(MAR∨MAE[;1])∧MAE[;2]
[11]  ⍝ MAE REPRESENTS THE FALUTS PLACED INTO THE MAR.
[12]  ⍝ DEE REPRESENTS THE FAULTS PLACED INTO THE DECODER.
[13]  TM1←((∼MAR)∨∨/DEE[⍳5; 1 2 3])∧DEE[⍳5;4]
[14]  ⍝   1  2  3  4  5  6  7  8  9 10
[15]  ⍝  ×4 ×̄4 ×3 ×̄3 ×2 ×̄2 ×1 ×̄1 ×0 ×̄0
[16]  ⍝ THE ABOVE TWO LINES REPRESENT THE VECTOR D WHICH IS D
      EFINED NEXT.
[17]  D←((10⍴ 1 0)\MAR)∨(10⍴ 0 1)\TM1
[18]  ⍝ TM2 REPRESENTS INPUTS TO THE FIRST LEVEL OF THE DECOD
      ER WITH OUT FAULTS.
[19]  TM2← 4 2 ⍴D[7 9 7 10 8 9 8 10]
[20]  ⍝ DI2 REPRESENTS THE INPUTS TO THE FIRST LEVEL OF THE D
      ECODER WITH FAULTS.
[21]  DI2←TM2∨DEE[I←5+⍳4; 1 2]
[22]  ⍝ D2 REPRESENTS THE OUTPUT OF THE FIRST LEVEL OF THE DE
      CODER WITH FALUTS.
[23]  D2←((,DEE[I;3])∨∧/DI2)∧,DEE[I;4]
[24]  ⍝                        ----- THE SECTION OF THE ABOVE EXPRESSI
      ON (INDICATED BY THE UNDERLINE) REPRESENTS THE
[25]  ⍝                              OUTPUT OF THE FIRST LEVEL. EG.(×1
      ∧×0) (×1∧×̄0) (×̄1∧×0) (×̄1∧×̄0)
[26]  ⍝ TM3 REPRESENTSTHE INPUTS TO THE SECOND LEVEL OF THE D
      ECODER.
[27]  TM3← 8 2 ⍴((16⍴ 0 1)\D2,D2)∨(8⍴D[5],0),8⍴D[6],0
[28]  DI3←TM3∨DEE[I←9+⍳8; 1 2]
[29]  D3←((,DEE[I;3])∨∧/DI3)∧,DEE[I;4]
[30]  TM4← 16 2 ⍴((32⍴ 0 1)\D3,D3)∨(16⍴D[3],0),16⍴D[
      4],0
[31]  DI4←TM4∨DEE[I←17+⍳16; 1 2]
[32]  D4←((,DEE[I;3])∨∧/DI4)∧,DEE[I;4]
[33]  TM5← 32 2 ⍴((64⍴ 0 1)\(D4,D4))∨(32⍴D[1],0),32⍴D[
      2],0
[34]  DI5←TM5∨DEE[I←33+⍳32; 1 2]
[35]  D5←((,DEE[I;3])∨∧/DI5)∧,DEE[I;4]
[36]  INX←D5/D5×⍳32
[37]  →OPEN×⍳0=+/D5
[38]  →RAD×⍳RDWR=1
[39]  MEM[INX]←IN
[40]  MEM←(MEE[;1]∨MEM)∧MEE[;2]
[41]  →LOOP
[42] RAD:OUTPUT←OUTPUT,∨/MEM[INX]
[43]  NUMACS←NUMACS,(0=⍴INX)+(0≠⍴INX)×INX
[44]  STKINX←STKINX,INX
[45]  →LOOP
[46] OPEN:OUTPUT←OUTPUT,0
[47]  NUMACS←NUMACS,0
[48]  →LOOP
    ∇
```

```
     ∇ATS [⎕]∇
   ∇ ATS
[1]    ⍝THIS PROGRAM REPRESENTS AN OPTIMAL ALGORITHM FOR TESTI
       NG RANDOM ACCESS MEMORIES.
[2]    PI0←(32⍴ 1 0 0)/IX←¯1+⍳32
[3]    PI1←(32⍴ 0 1 0)/IX
[4]    PI2←(32⍴ 0 0 1)/IX
[5]    ⍝_______________________________________________STEP 1
[6]    RDWR←IN←0
[7]    MEMORY PI1
[8]    MEMORY PI2
[9]    ⍝_______________________________________________STEP 2
[10]   IN←1
[11]   MEMORY PI0
[12]   ⍝_______________________________________________STEP 3
[13]   IN←0
[14]   RDWR←1
[15]   MEMORY PI1
[16]   3 OUT PI1
[17]   ⍝_______________________________________________STEP 4
[18]   IN←1
[19]   RDWR←0
[20]   MEMORY PI1
[21]   ⍝_______________________________________________STEP 5
[22]   IN←0
[23]   RDWR←1
[24]   MEMORY PI2
[25]   5 OUT PI2
[26]   ⍝_______________________________________________STEP 6
[27]   IN←1
[28]   MEMORY PI0
[29]   6 OUT PI0
[30]   ⍝_______________________________________________STEP 7
[31]   IN←RDWR←0
[32]   MEMORY PI0
[33]   ⍝_______________________________________________STEP 8
[34]   IN←1
[35]   MEMORY PI2
[36]   ⍝_______________________________________________STEP 9
[37]   IN←0
[38]   RDWR←1
[39]   MEMORY PI0
[40]   9 OUT PI0
[41]   ⍝_______________________________________________STEP 10
[42]   IN←1
[43]   MEMORY PI2
[44]   10 OUT PI2
[45]   ⍝_______________________________________________STEP 11
[46]   MEMORY PI1
[47]   11 OUT PI1
   ∇
```

```
    ∇O̲U̲T̲ [⎕]∇
    ∇ STEP O̲U̲T̲ PI
[1]   ⍝THIS PROGRAM IS TO BE USED WITH THE 'ATS' PROGRAM TO O
      UTPUT THE DETECTED FAILURES;
[2]   →0×⍳1>+/TEM←I̲N̲≠OUTPUT
[3]   INX←TEM/TEM×⍳⍴PI
[4]   'STEP ';STEP
[5]   'THE FOLLOWING LOCATIONS READ THE ALL ';∼I̲N̲;' WORD INST
      EAD OF THE ALL ';I̲N̲;' WORD;'
[6]   PI[INX]
    ∇
```

```
    ∇PRINT  [⎕]∇
   ∇ PRINT CHOICE
[1]    ⍝ THIS PROGRAM WILL OUTPUT THE INDUCED ERRORS PLACED IN
      TO THE DECODER.
[2]    →(CHOICE-1)⌽ST1,ST1,ST1,ST2
[3]   ST1:I̲M̲2←G[;(¯4 ¯3 ¯2 ¯1 +5×(,I̲M̲1)/(,I̲M̲1)×⍳65)]
[4]    X1←(∼' '=,I̲M̲2)/,I̲M̲2
[5]    →(CHOICE-1)⌽PR1,PR2,PR3
[6]   PR1:X2←(∼' '=,I̲M̲2[; 1 2])/,I̲M̲2[; 1 2]
[7]   END12:'      THE ';X2;' INPUT TO GATE ';X1;' IS STUCK-AT-
      ONE.'
[8]    →0
[9]   PR2:X2←(∼' '=,I̲M̲2[; 3 4])/,I̲M̲2[; 3 4]
[10]   →END12
[11] PR3:'      THE OUTPUTOF GATE ';X1;' IS STUCK-AT-ONE.'
[12] →0
[13] ST2:I̲M̲2←G[;(¯4 ¯3 ¯2 ¯1 +5×(∼,I̲M̲1)/(∼,I̲M̲1)×⍳65)]
[14]  X1←(∼' '=,I̲M̲2)/,I̲M̲2
[15]  '      THE OUTPUT OF GATE ';X1;' IS STUCK-AT-ZERO.'
   ∇
```

```
     ∇AUTODEC [⎕]∇
    ∇ AUTODEC CHOOSE
[1]   ⍝THIS PROGRAM WILL AUTOMATICLY GENERATE ALL SINGLE FAIL
      URES IN THE DECODER OF A 32×1BIT RAM.
[2]   ⍝CHOOSE=1 WILL PLACE THE INPUT ON ONE SIDE OF ALL 'AND'
       GATES TO STUCK-AT-ONE.
[3]   ⍝CHOOSE=2 WILL PLACE THE INPUT ON THE OTHER SIDE OF ALL
       'AND' GATES TO STUCK-AT-ONE.
[4]   ⍝CHOOSE=3 WILL PLACE THE OUTPUT OF ALL 'AND' GATES TO S
      TUCK-AT-ONE.
[5]   ⍝CHOOSE=4 WILL PLACE THE OUTPUT OF ALL 'AND' GATES TO S
      TUCK-AT-ZERO.
[6]   ⍝THE OUTPUT OF THIS PROGRAM WILL SHOW THE OUTPUT OF THE
       ATS FOR THE DETECTED FAILURES.
[7]   CLEARERRORS
[8]   →ZERO×⍳4=CHOOSE
[9]   I̲M̲1← 65 1 ⍴1,64⍴0
[10] LOOP:PRINT CHOOSE
[11]  DEE[;CHOOSE]←I̲M̲1
[12]  ATS
[13]  →0×⍳1= 1 1 ↑I̲M̲1←¯1⊖I̲M̲1
[14]  →LOOP
[15] ZERO:I̲M̲1← 65 1 ⍴0,64⍴1
[16] LOOP2:PRINT CHOOSE
[17]  DEE[;4]←I̲M̲1
[18]  ATS
[19]  →0×⍳0= 1 1 ↑I̲M̲1←¯1⊖I̲M̲1
[20]  →LOOP2
    ∇




     ∇CLEARERRORS [⎕]∇
    ∇ CLEARERRORS
[1]   ⍝THIS PROGRAM WILL CLEAR ALL ERRORS PREVIOUSLY SET IN ALL SUBSY
      STEMS OF THE 32×1 BIT RAM.
[2]   MEE←⍉ 2 32 ⍴(32⍴0),32⍴1
[3]   MEM←32⍴0
[4]   MAE←⍉ 2 5 ⍴(5⍴0),5⍴1
[5]   DEE←⍉ 4 65 ⍴(195⍴0),65⍴1
    ∇
```

```
        AUTODEC 4
       THE OUTPUT OF GATE X̲4 IS STUCK-AT-ZERO.
STEP 6
THE FOLLOWING LOCATIONS READ THE ALL 0 WORD INSTEAD OF THE ALL
        1 WORD:
0  3  6  9  12  15
STEP 10
THE FOLLOWING LOCATIONS READ THE ALL 0 WORD INSTEAD OF THE ALL
        1 WORD:
2  5  8  11  14
STEP 11
THE FOLLOWING LOCATIONS READ THE ALL 0 WORD INSTEAD OF THE ALL
        1 WORD:
1  4  7  10  13
      THE OUTPUT OF GATE X̲3 IS STUCK-AT-ZERO.
STEP 6
THE FOLLOWING LOCATIONS READ THE ALL 0 WORD INSTEAD OF THE ALL
        1 WORD:
0  3  6  18  21
STEP 10
THE FOLLOWING LOCATIONS READ THE ALL 0 WORD INSTEAD OF THE ALL
        1 WORD:
2  5  17  20  23
STEP 11
THE FOLLOWING LOCATIONS READ THE ALL 0 WORD INSTEAD OF THE ALL
        1 WORD:
1  4  7  16  19  22
      THE OUTPUT OF GATE X̲2 IS STUCK-AT-ZERO.
```

```
     ∇AUTOMAR [⎕]∇
    ∇ AUTOMAR
[1]    ⍝ THIS PROGRAM WILL SIMULATE ALL SINGLE STUCK-AT FAULTS
        IN THE MAR.
[2]    CLEARERRORS
[3]    JMA←IMA←0
[4]    KMA←W̲←1
[5]   LF:→RES×⍳5≥JMA←JMA+1
[6]    JMA←1
[7]   RES:→SET×⍳5≥IMA←IMA+1
[8]    W̲←0
[9]    KMA←2
[10]  SET:''
[11]   ''
[12]   '                 MAR BIT ';5|(JMA×9);' IS S-A-';W̲
[13]   MAE← 5 2 ⍴ 0 1
[14]   MAE[JMA;KMA]←W̲
[15]   ATS
[16]   →LF×⍳10>IMA
    ∇
```

```
     ∇SETERROR [⎕]∇
   ∇ SETERROR
[1]    ⍝ THIS PROGRAM IS USED TO SET THE 'STUCK-AT' ERRORS IN
       THE 32X1 BIT MEMORY DESCRIBED IN THE PROGRAM 'MEMORY'.
[2]    ' TO EXIT ENTER ; EXIT .

       '
[3]   STA:'    ENTER THE MEMORY SUBSYSTEM THAT YOU WISH TO INTR
       ODUCE THE ERROR INTO.     '
[4]    'THAT IS: MAR ; DEC ; MEM .'
[5]    'TO CHOOSE ANOTHER SUBSYSTEM ENTER: START
[6]    →0×⍳∧/'EXI'=BCH←3⍴⍞,'EXI'
[7]    →DEC×⍳∧/'DEC'=BCH
[8]    →MEM×⍳∧/'MEM'=BCH
[9]   MAR:'INDICATE WHICH MAR BITS ARE FAULTY BY SIGNIFING A N
       UMERIC 0 FOR A S-A-0 AND A NUMERIC 1 FOR A S-A-1'
[10]   'UNDER THE APPROPRIATE BIT POSITION.'
[11]   ' ×4 | ×3 | ×2 | ×1 | ×0 |'
[12]   TEM← 5 5 ⍴⍞,25⍴' '
[13]   MAE[;1]←∨/'1'=TEM
[14]   MAE[;2]←~∨/'0'=TEM
[15]   →STA
[16] DEC:'INDICATE A S-A-0 BY PLACING AT LEAST ONE NUMERIC 0
       BELOW THE INDICATED GATE INPUTS.'
[17]   'IN ORDER TO INDICATE A S-A-1 ON A GATE OUTPUT,PLACE TW
       O NUMERIC 1''S UNDER THE SPECIFIED GATE.'
[18]   'IF JUST A SINGLE S-A-1 ON A GATE IS REQUIRED, PLACE A
       NUMERIC 1 UNDER THE SPECIFIC GATE INPUT.'
[19]   'TO EXIT ENTER: EXIT '
[20]   'TO RETURN TO THE START OF THIS PROGRAM ENTER: START '
[21]   GA←(4 25 ⍴'    |'),[1] 1 25 ⍴'  ×̲4|  ×̲3|  ×̲2|  ×̲1|  ×̲0|
       '
[22]   GB←(300⍴'  ×0|  ×̲0|'),'×1  |×1  |×̲1  |×̲1  |',(
       280⍴'  ×1|  ×1|  ×̲1|  ×̲1|')
[23]   GC←(20⍴'    |'),(40⍴(20⍴'×2  |'),20⍴'×̲2  |'),(
       240⍴(20⍴'  ×2|'),20⍴'  ×̲2|')
[24]   GD←(60⍴'    |'),(80⍴(40⍴'×3  |'),40⍴'×̲3  |'),(
       160⍴(40⍴'  ×3|'),40⍴'  ×̲3|')
[25]   GE←(140⍴'    |'),160⍴(80⍴'×4  |'),80⍴'×̲4  |'
[26]   G←GA,⍉ 5 300 ⍴GB,GC,GD,GE
[27]   GA←GB←GC←GD←GE←⍴0
[28]   'THE FIRST SET OF GATE ARE:'
[29]   5 120 ↑G
[30]   TEM←120⍴⍞,120⍴' '
[31]   5 ¯120 ↑ 5 240 ↑G
[32]   TEM←TEM,120⍴⍞,120⍴' '
[33]   5 ¯85 ↑G
[34]   TEM← 65 5 ⍴TEM,85⍴⍞,85⍴' '
[35]   DEE[;4]← 65 1 ⍴~∨/'0'=TEM
[36]   DEE[;3]←(65 1 ⍴DEE[;4])∧TM1← 65 1 ⍴2=+/'1'=TEM
[37]   DEE[;1]←(~TM1)∧ 65 1 ⍴∨/'1'=TEM[; 1 2]
[38]   DEE[;2]←(~TM1)∧ 65 1 ⍴∨/'1'=TEM[; 3 4]
[39]   →STA
[40] MEM:'INDICATE WHICH STORAGE WORDS ARE FAULTY BY SIGNIFIN
       G A NUMERIC 0 FOR A S-A-0 AND A NUMERIC 1'
[41]   'FOR A S-A-1 UNDER THE APPROPRIATE STORAGE ADDRESS.'
[42]   ' 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|16|17|
       18|19|20|21|22|23|24|25|26|27|28|29|30|31|'
[43]   TEM← 32 3 ⍴⍞,96⍴' '
[44]   MEE[;2]← 32 1 ⍴~∨/'0'=TEM
[45]   MEE[;1]← 32 1 ⍴∨/'1'=TEM
[46]   →STA
   ∇
```

```
       SETERROR
 TO EXIT ENTER ; EXIT .


    ENTER THE MEMORY SUBSYSTEM THAT YOU WISH TO INTRODUCE THE ERROR INTO,
THAT IS; MAR ; DEC ; MEM .
TO CHOOSE ANOTHER SUBSYSTEM ENTER; START     .
MAR
INDICATE WHICH MAR BITS ARE FAULTY BY SIGNIFING A NUMERIC 0 FOR A S-A-0 AND A NUMERIC 1 FOR A S-A-1
UNDER THE APPROPRIATE BIT POSITION,
 X4 | X3 | X2 | X1 | X0 |
0          0           1
    ENTER THE MEMORY SUBSYSTEM THAT YOU WISH TO INTRODUCE THE ERROR INTO,
THAT IS; MAR ; DEC ; MEM .
TO CHOOSE ANOTHER SUBSYSTEM ENTER; START     .
DEC
INDICATE A S-A-0 BY PLACING AT LEAST ONE NUMERIC 0 BELOW THE INDICATED GATE INPUTS,
IN ORDER TO INDICATE A S-A-1 ON A GATE OUTPUT,PLACE TWO NUMERIC 1'S UNDER THE SPECIFIED GATE,
IF JUST A SINGLE S-A-1 ON A GATE IS RZQUIRED, PLACE A NUMERIC 1 UNDER THE SPECIFIC GATE INPUT,
TO EXIT ENTER; EXIT
TO RETURN TO THE START OF THIS PROGRAM ENTER; START
THE FIRST SET OF GATE ARE;
      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |X3    |X3    |X3    |X3    |X3    |X3    |X3    |
      |      |      |      |      |      |      |      |      |X2    |X2    |X2    |X2    |X̲2    |X̲2    |X̲2    |X̲2    |    X2|    X2|    X2|    X2|    X̲2|    X̲2|    X̲2|
      |      |      |      |      |X1    |X1    |X̲1    |X̲1    |    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|
    X̲4|    X̲3|    X̲2|    X̲1|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|   .X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|
          0                0          1       1      11              0                                         11          0
      |      :      |      |      |      |      |      |      |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |X4    |
X3     |X̲3    |X̲3    |X̲3    |X̲3    |X̲3    |X̲3    |X̲3    |X̲3    |    X3|    X3|    X3|    X3|    X3|    X3|    X3|    X3|    X̲3|    X̲3|    X̲3|    X̲3|    X̲3|    X̲3|    X̲3|
    X̲2|    X2|    X2|    X2|    X2|    X̲2|    X̲2|    X̲2|    X̲2|    X2|    X2|    X2|    X2|    X̲2|    X̲2|    X̲2|    X̲2|    X2|    X2|    X2|    X2|    X̲2|    X̲2|    X̲2|
    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1'    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|
    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|   .X0'    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|
                                                                           0                                           1
X4     |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |X̲4    |
    X̲3|    X3|    X3|    X3|    X3|    X3|    X3|    X3|    X3|    X̲3|    X̲3|    X̲3|    X̲3|    X̲3|    X̲3|    X̲3|    X̲3|
    X̲2|    X2|    X2|    X2|    X2|    X̲2|    X̲2|    X̲2|    X̲2|    X2|    X2|    X2|    X2|    X̲2|    X̲2|    X̲2|    X̲2|
    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|    X1|    X1|    X̲1|    X̲1|
    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|    X0|    X̲0|
                                                                                                       0
    ENTER THE MEMORY SUBSYSTEM THAT YOU WISH TO INTRODUCE THE ERROR INTO,
THAT IS; MAR ; DEC ; MEM .
TO CHOOSE ANOTHER SUBSYSTEM ENTER; START     .
MEM
INDICATE WHICH STORAGE WORDS ARE FAULTY BY SIGNIFING A NUMERIC 0 FOR A S-A-0 AND A NUMERIC 1
FOR A S-A-1 UNDER THE APPROPRIATE STORAGE ADDRESS,
 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31|
       1           1     0  0  0                                                  1  0  0

    ENTER THE MEMORY SUBSYSTEM THAT YOU WISH TO INTRODUCE THE ERROR INTO,
THAT IS; MAR ; DEC ; MEM .
TO CHOOSE ANOTHER SUBSYSTEM ENTER; START     .
EXIT
```

References

(1) Knaizuk, John Jr., Hartmann, C.R.P., "An Algorithm For Testing Random Access Memories", TR 74-14, Department of Industrial Engineering and Operations Research, Syracuse University.

(2) Knaizuk, John Jr., Hartmann, C.R.P., "An Optimal Algorithm for Testing Random Access Memories", TR 75-16, Department of Industrial Engineering and Operations Research, Syracuse University.

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared		m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	µ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* To be avoided where possible.